



TRƯỜNG CĐ CÔNG THƯƠNG TP. HỒ CHÍ MINH
KHOA: CÔNG NGHỆ THÔNG TIN



NHẬP MÔN LẬP TRÌNH



Ths: Huỳnh Nguyễn Thành Luân

Khoa Công Nghệ Thông Tin

Email: hntluan2005@gmail.com

Website thông tin môn học:

www.hntluan.com



Thông tin về môn học

❖ **Thời lượng:** 30 tiết lý thuyết + 30 tiết thực hành (3TC)

❖ **Nội dung môn học:**

- Giới thiệu các khái niệm cơ bản về lập trình.
- Cung cấp cơ sở lý thuyết và kỹ năng cơ bản về lập trình cho các môn học sau.
- Kỹ năng phân tích bài toán, viết được chương trình để giải bài toán. Có khả năng viết chương trình bằng nhiều cách khác nhau.

❖ **Đánh giá kết quả:**

- **Điểm quá trình:** 40%
 - + Phát biểu, làm các bài tập, bài tập nhóm và chuyên cần.
 - + Kiểm tra giữa kỳ.
- **Thi kết thúc:** 60%

Hình thức thi tự luận 90'

CÁC NỘI DUNG

Chương 1: Giới thiệu chung

Chương 2: Kiểu dữ liệu, hằng, biến, biểu thức

Chương 3: Các cấu trúc điều khiển

Chương 4: Hàm và cấu trúc chương trình

Chương 5: Mảng, chuỗi ký tự, con trỏ

Chương 6: Kiểu dữ liệu có cấu trúc (tự học)

Tài liệu học tập – tham khảo

❖ Giáo trình chính:

- Slide bài giảng
- GS. Phạm Văn Ất, **Giáo Trình Kỹ Thuật Lập Trình C - Căn Bản & Nâng Cao**, NXB Hồng Đức, 2009

❖ Tài liệu tham khảo:

- [1] Trần Đan Thư – Giáo trình lập trình C tập 1-2, NXB Đại Học Quốc Gia TP.HCM
- [2] Huỳnh Tấn Dũng & Hoàng Đức Hải, Bài tập ngôn ngữ C, NXB Lao động xã hội, 2006.
- [3] 2003.Nguyễn Đình Tê & Hoàng Đức Hải, Giáo trình lý thuyết và bài tập C, NXB Giáo dục, 2006
- [4] Lê Minh Hoàng, Giải thuật lập trình, ebook.
- [5] Trần Minh Thái, Giáo trình bài tập kỹ thuật lập trình, ebook.

Tài liệu học tập – tham khảo

❖ Phần mềm:

- **Borland C++ 5.02** hoặc tương đương **Dev-C++ 4.9.9.2**, **Visual Studio 2010** (C++ Win32 Console Application)
- Công cụ vẽ lưu đồ **Crocodile 6.05**

❖ Website học tập:

www.hntluan.com

www.fit-hitu.edu.vn/forum

www.congdongcviet.com

www.google.com.vn

Chương 1

Giới thiệu chung

- ❖ Một số khái niệm cơ bản
- ❖ Giải thuật và lưu đồ
- ❖ Giới thiệu về ngôn ngữ lập trình C
- ❖ Các thành phần cơ bản của ngôn ngữ lập trình C

Các khái niệm cơ bản

- ❖ **Chương trình (program):** Một đoạn mã lệnh
- ❖ **Lập trình (programming):** viết chương trình cho máy tính (coding)
- ❖ **Ngôn ngữ lập trình (programming language)**
 - Phương tiện để viết chương trình cho máy tính
 - Hàng trăm ngôn ngữ lập trình khác nhau
 - Những quy định về cú pháp (syntax) & ngữ nghĩa (semantic)
 - Máy tính có thể hiểu được

Ngôn ngữ lập trình

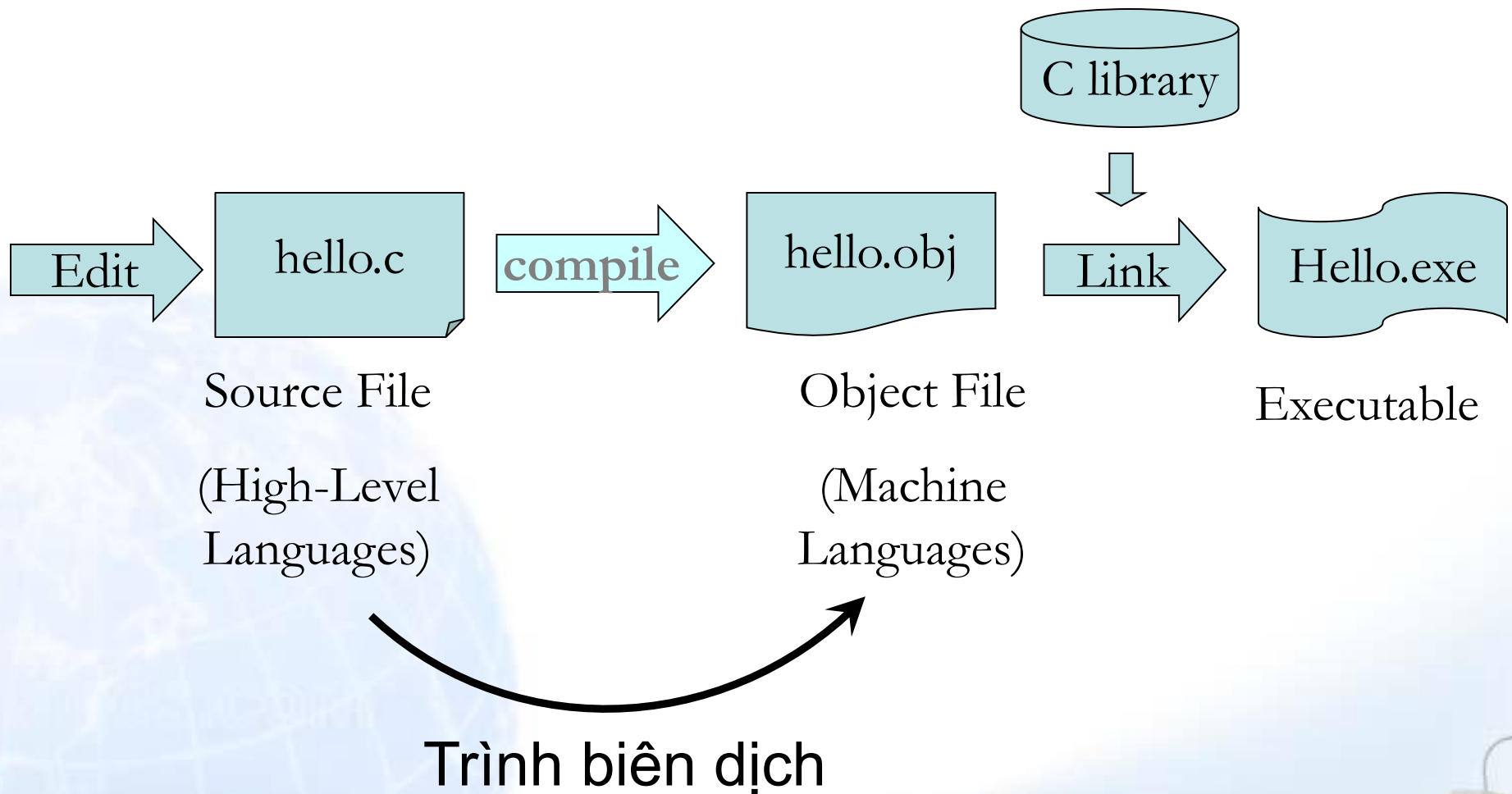
❖ Phân chia làm 3 nhóm chính

- *Ngôn ngữ máy - Machine languages*
 - Ngôn ngữ duy nhất của máy tính – CPU hiểu trực tiếp
 - Nền tảng xây dựng hợp ngữ
- *Hợp ngữ - Assembly languages*
 - Sử dụng các từ khóa tiếng Anh cho các lệnh hay nhóm lệnh của mã máy
 - Nền tảng xây dựng các ngôn ngữ cấp cao
- *Ngôn ngữ cấp cao - High-level languages*
 - Gần ngôn ngữ tự nhiên
 - Quy tắc ngữ pháp, Từ vựng, Cấu trúc câu

Chương trình dịch

- ❖ Dùng để dịch từ một ngôn ngữ lập trình này sang ngôn ngữ lập trình khác
- ❖ Mục tiêu cuối cùng là dịch sang mã máy để có được executed code → chương trình thực thi
- ❖ Phân loại:
 - **Interpreter – thông dịch:** Dịch từng lệnh một, dịch tới đâu thực hiện tới đó
 - **Compiler – biên dịch:** Dịch toàn bộ chương trình nguồn thành chương trình đích rồi sau đó mới thực hiện
- ❖ Công cụ phát triển – Integrated Development Environment (IDE)
 - Một phần mềm cung cấp toàn bộ những tiện ích cần thiết cho lập trình viên lập trình
 - Soạn thảo, dịch chương trình, chạy thử và sửa lỗi

Soạn thảo mã nguồn – Biên dịch – Liên kết và thực thi



Các ngôn ngữ lập trình

❖ Fortran

❖ Pascal

❖ C

❖ C++

❖ C#

❖ Visual Basic

❖ Java

❖

Các môi trường hỗ trợ lập trình (IDE)

- ❖ Borland C++
- ❖ Microsoft Visual Basic
- ❖ Microsoft Visual C++
- ❖ Dev-C++
- ❖ Jbuidar
- ❖ Eclipse SDK
- ❖ Microsoft Visual Studio
- ❖ ...

Một số khái niệm khác

❖ Lỗi và sửa lỗi

- *Syntax error – lỗi ngữ pháp*
- *Semantic error- lỗi ngữ nghĩa*
- *Runtime error - Lỗi thực thi*

❖ Debug – Tìm và sửa lỗi

❖ Giải thuật: *khái niệm, công cụ biểu diễn*

❖ Flow chart – lưu đồ

Giải thuật (algorithm)

❖ Định nghĩa

Giải thuật là tập hữu hạn các bước (các mã lệnh, các câu lệnh hoặc các chỉ thị) nhằm thực thi một yêu cầu nào đó, trong một khoảng thời gian nhất định và sau khi thực hiện hoàn tất giải thuật mang lại một kết quả mong muốn.

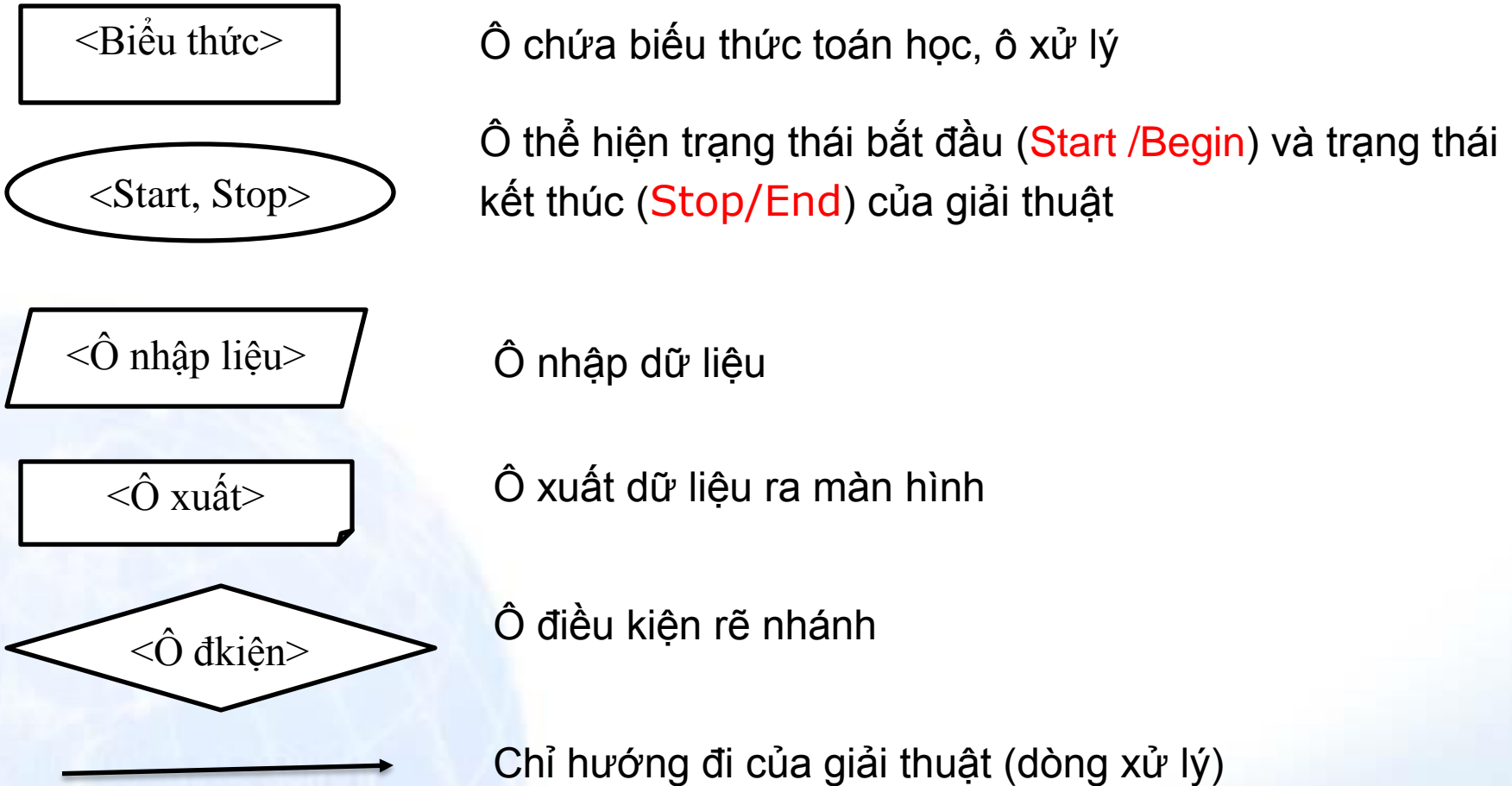
❖ Cách thức mô tả giải thuật

Có 4 cách thức thường dùng để mô tả giải thuật:

- *Mô tả bằng ngôn ngữ tự nhiên.*
- *Mô tả bằng mã giả.*
- *Mô tả bằng ngôn ngữ lập trình.*
- *Mô tả bằng lưu đồ.*

Giải thuật (algorithm)

Mô tả giải thuật bằng lưu đồ



Flow chart

❖ Ưu điểm

- Trình bày trực quan giải thuật
- Độc lập với ngôn ngữ tự nhiên
- Độc lập với ngôn ngữ lập trình
- Bảo đảm khả năng lập trình
- Cho phép dễ dàng kiểm tra giải thuật

❖ Nguyên tắc kiểm tra

- Đi từ **START** theo bất cứ đường nào cũng phải đến một điểm dừng **STOP**
- Không có sự quay vòng vĩnh viễn
- Không có sự kết thúc lưng chừng

Giải thuật (algorithm)

Để giải một bài toán chúng ta thực hiện (tư duy) theo 4 bước sau:

- + Đọc và hiểu bài toán.
- + Xác định các tham số đầu vào (Input) và tham số đầu ra (Output).
- + Vẽ mô hình giải thuật cho bài toán.
- + Viết chương trình dựa vào mô hình giải thuật.

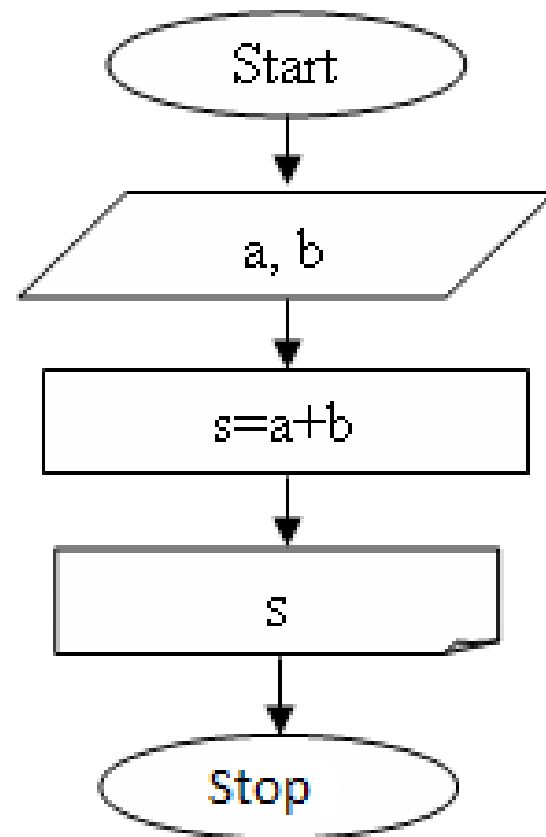
Các ví dụ:

Giải thuật (algorithm)

Nhập vào 2 số nguyên bất kỳ a, b . Tính tổng của chúng rồi xuất ra màn hình.

Input: a, b

Output: s

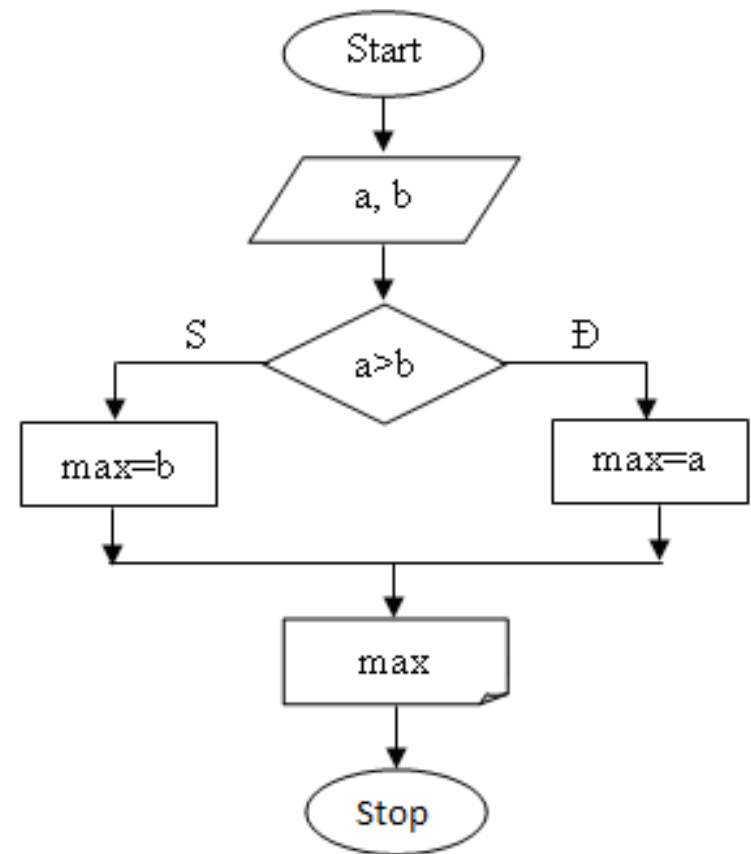


Giải thuật (algorithm)

Nhập vào 2 số nguyên bất kỳ a, b . Tìm số lớn nhất rồi in ra màn hình.

Input: a, b

Output: \max



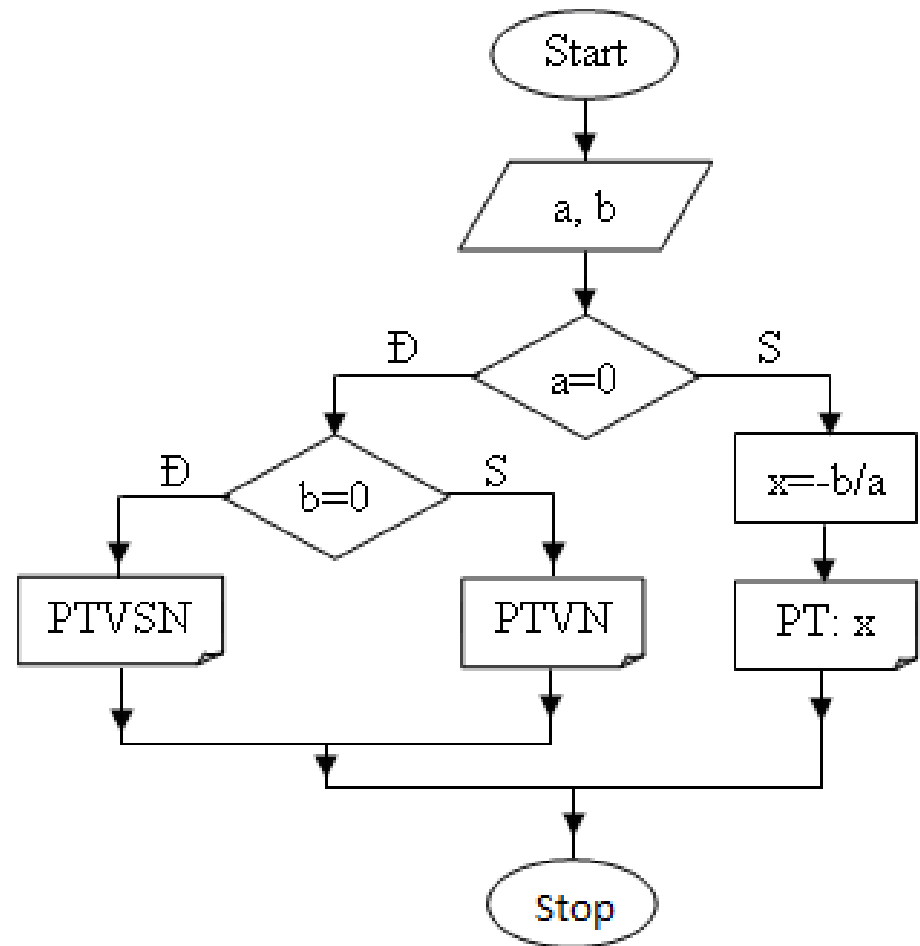
Giải thuật (algorithm)

Giải phương trình bậc 1:

$$ax + b = 0$$

Input: a, b

Output: x , PTVN,
PTVSN

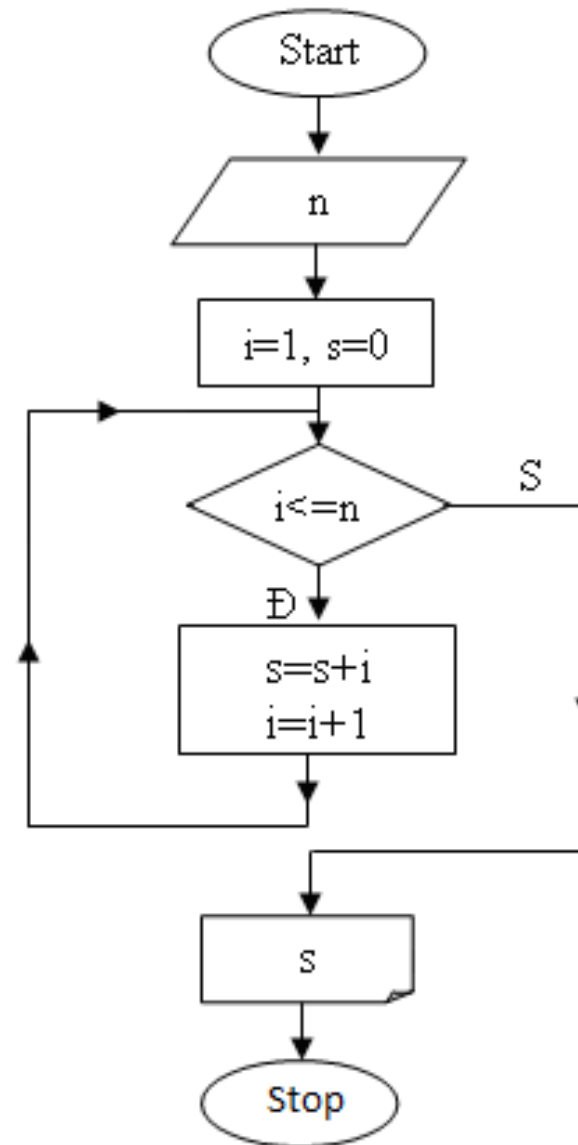


Giải thuật (algorithm)

➤ Nhập vào số nguyên n bất kỳ. Tính tổng từ 1 đến n rồi in ra màn hình.

➤ Input: n

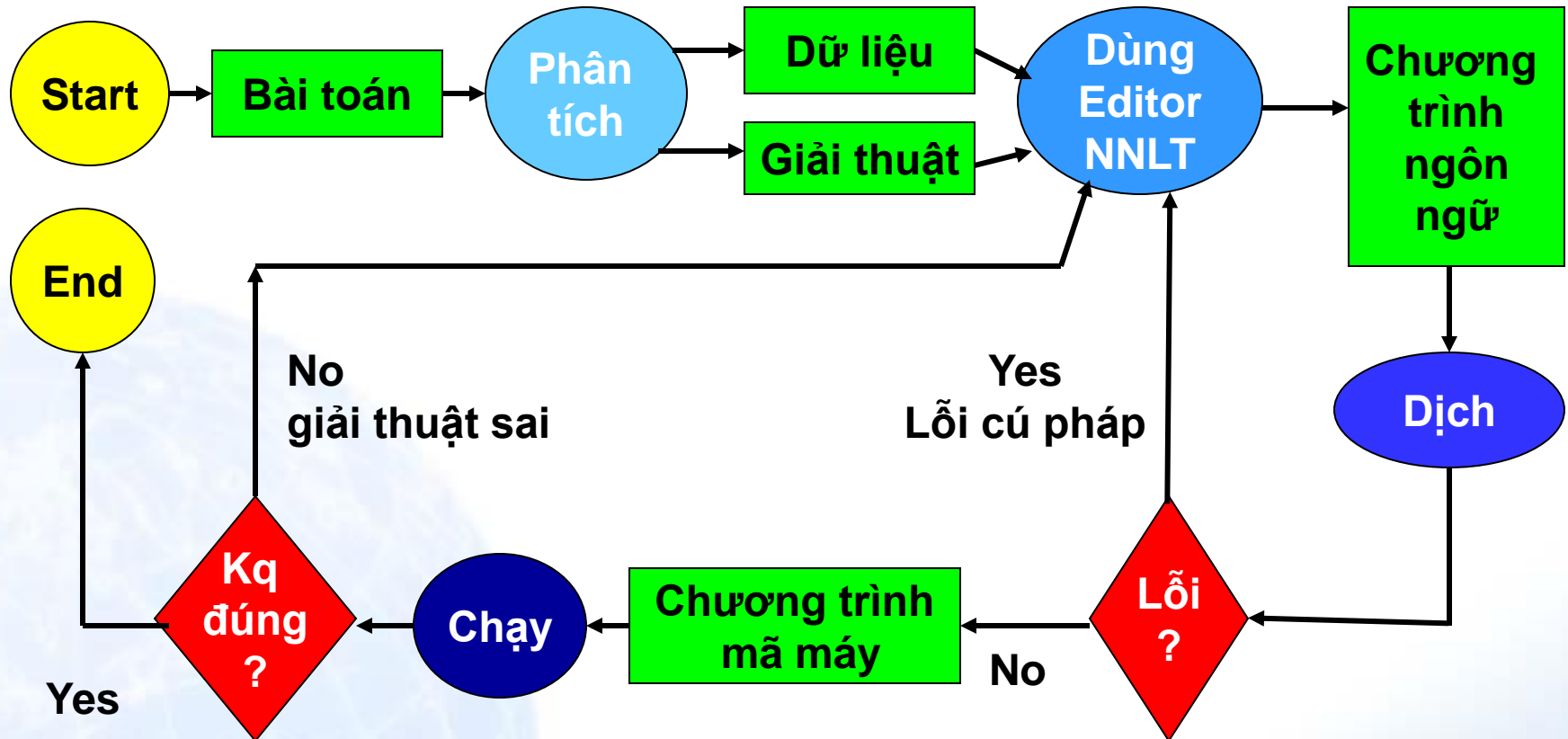
➤ Output: $s=1+2+....+n$



CÁC VÍ DỤ

1. Cho số nguyên n . Tính trị tuyệt đối của n
2. Nhập vào số nguyên k ($k > 0$), Xuất ra màn hình k dòng chữ “Xin chào”
3. Tính tổng: $S = 1 + 3 + 5 + \dots + (2n + 1)$, với $n \geq 0$
4. Tính tổng: $S(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1}n$, với $n > 0$
5. Nhập vào ba cạnh a, b, c của tam giác. Xuất ra màn hình tam giác đó thuộc loại tam giác gì?
(*Thường, cân, vuông, đều hay vuông cân*).

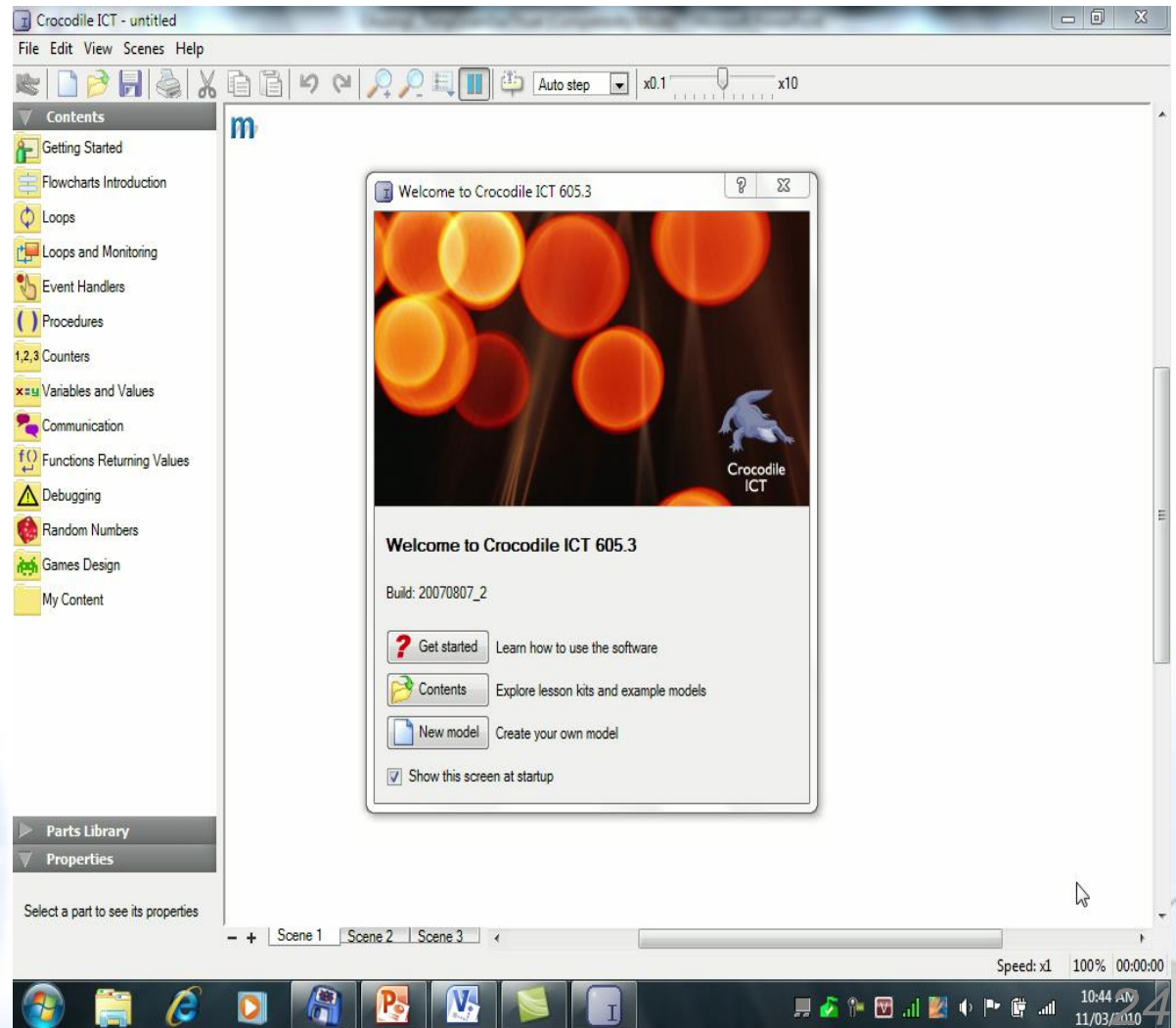
Các bước lập trình



Công cụ vẽ lưu đồ giải thuật

❖ Microsoft Visio

❖ Crocodile 6.05



Tổng quan về ngôn ngữ C

- ❖ Ngôn ngữ C do Dennis Ritchie xây dựng từ năm 1972 tại phòng thí nghiệm Bell Telephone với mục đích tạo ngôn ngữ để viết HĐH UNIX.
- ❖ C dựa trên các ngôn ngữ đã có:
 - *ALGOL 60 (1960),*
 - *CPL (Cambridge, 1963),*
 - *BCPL (Martin Richard, 1967),*
 - *B (Ken Thompson, 1970)*
- ❖ C là ngôn ngữ lập trình cấp cao, được sử dụng rất phổ biến để lập trình hệ thống cùng với Assembler và phát triển các ứng dụng
- ❖ Năm 1978, xuất bản quyển sách “The C programming language” do Kernighan và Ritchie viết.

Những đặc điểm của C

❖ *Tính cô đọng (compact):*

- *C chỉ có 32 từ khóa chuẩn và*
- *40 toán tử chuẩn.*

❖ *Tính cấu trúc (structured):*

- *C có một tập hợp những chỉ thị lập trình: cấu trúc lựa chọn, lặp, ...*
- *Đơn giản dễ hiểu.*

❖ *Tính tương thích (compatible):*

- *C có bộ tiền xử lý và*
- *Các thư viện chuẩn vô cùng phong phú*
- *Nên khi chuyển từ máy tính này sang máy tính khác các chương trình viết bằng C vẫn hoàn toàn tương thích.*

Những đặc điểm của C (tt)

❖ *Tính linh động (flexible):*

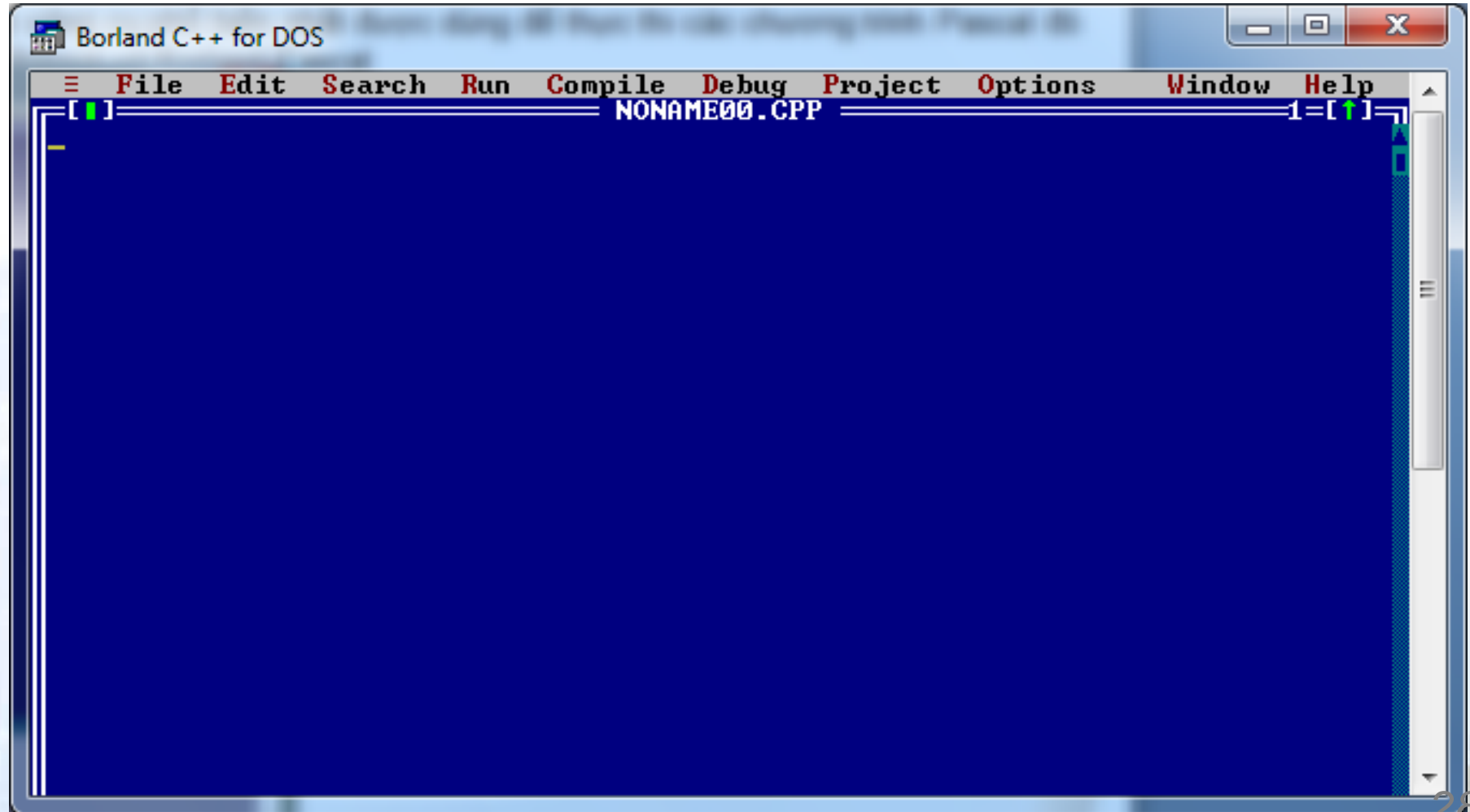
- *Cú pháp rất uyển chuyển, chấp nhận nhiều cách thể hiện*
- *Có thể thu gọn kích thước của mã lệnh*
- *Làm chương trình chạy nhanh hơn*

❖ *Biên dịch (compile):*

- *C cho phép biên dịch nhiều tập tin chương trình riêng rẽ thành các tập tin đối tượng (object) và*
- *Liên kết (link) các đối tượng đó lại với nhau thành một chương trình có thể thực thi được (executable) thống nhất*

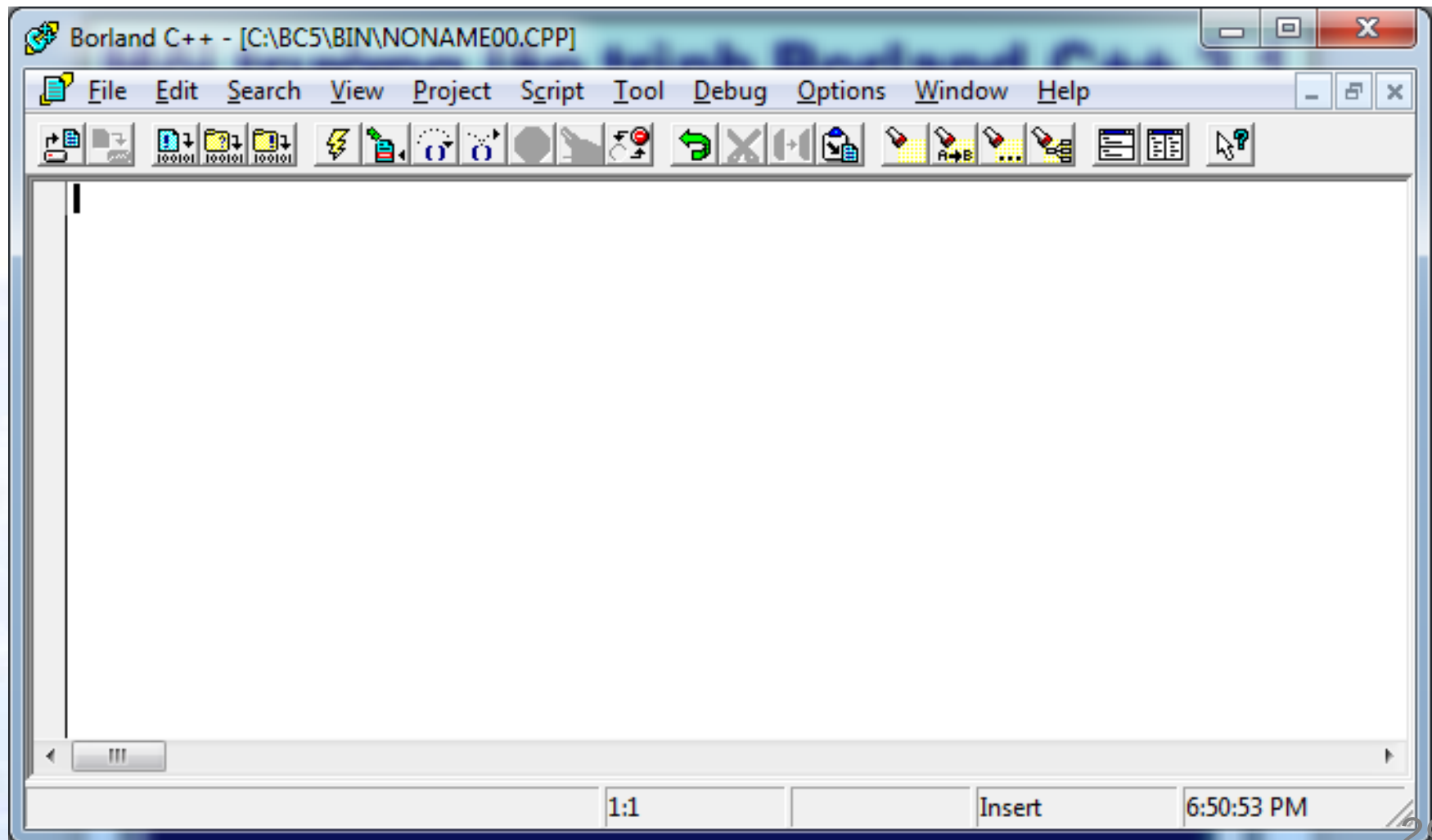
Môi trường lập trình Borland C++ 3.1

Ta có thể khởi động **Borland C++ 3.1** từ Windows hoặc MS-DOS chạy tập tin BC.EXE



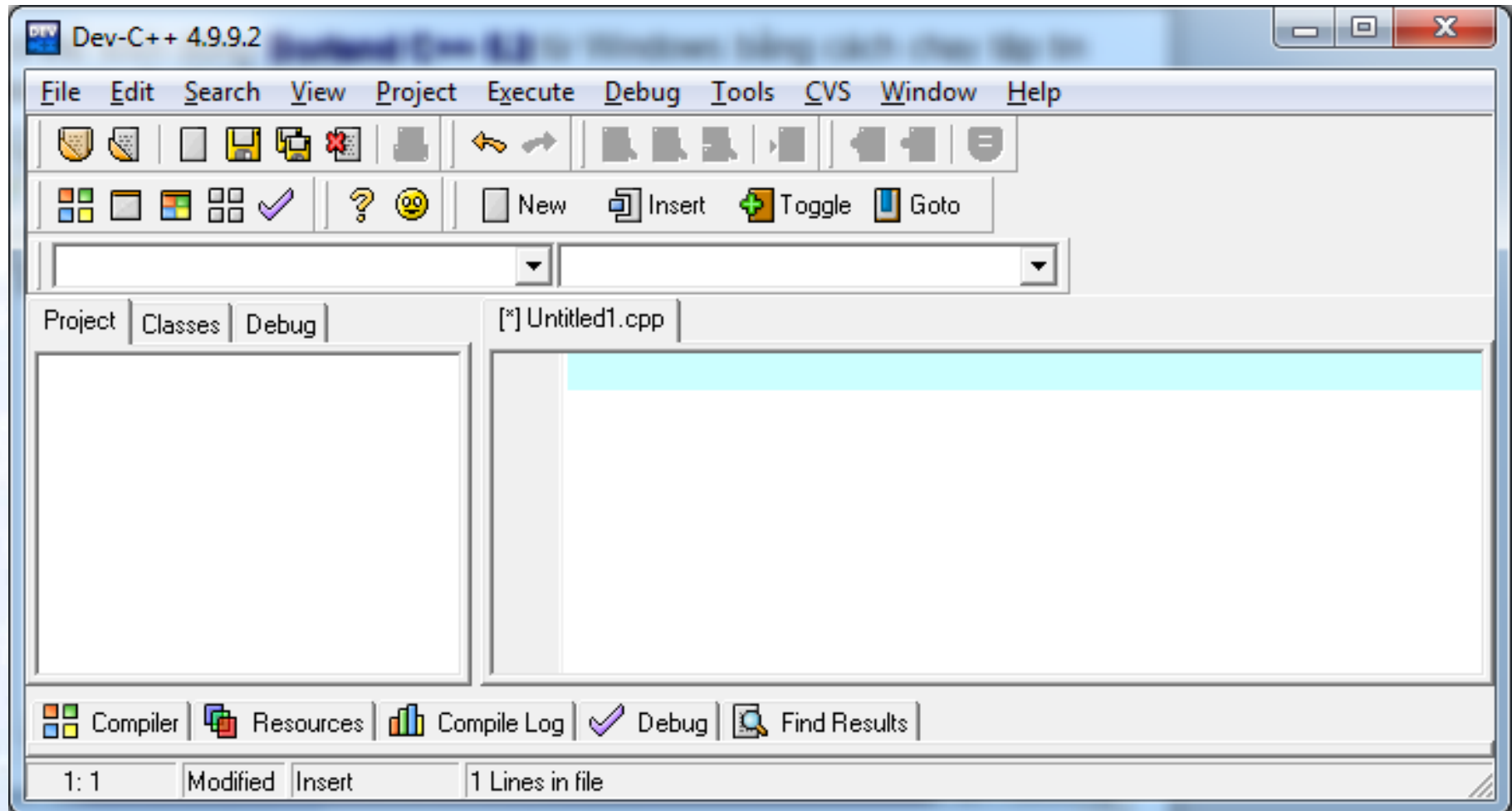
Môi trường lập trình Borland C++ 5.2

Ta có thể khởi động **Borland C++ 5.2** từ Windows bằng cách chạy tập tin **bcw.exe** từ **C:\BC5\BIN\bcw.exe**

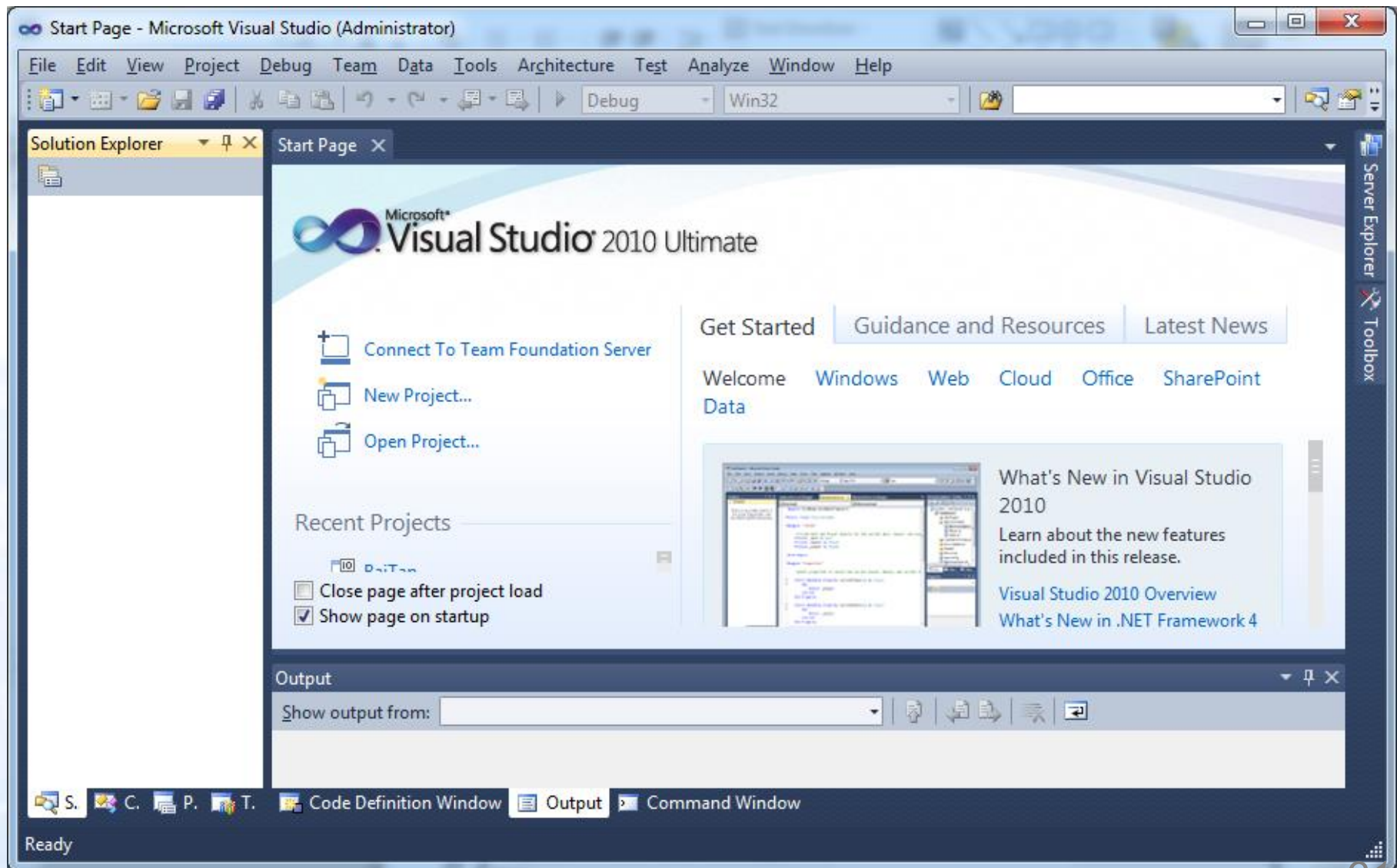


Môi trường lập trình Dev-C++ 4.9.9.2

Khi cài đặt mặc định C:\Dev-Cpp\devcpp.exe



Môi trường lập trình Visual Studio 2010



Môi trường lập trình Borland C++

Các phím chức năng cơ bản cần biết trong **Borland C++ 5.02**

Phím tắt	Ý nghĩa
Alt + F9	Dịch chương trình
F9	Make all command - Project
Ctrl + F9	Dịch và chạy chương trình.
Ctrl + Delete	Clear command - Edit

Môi trường Visual Studio

Các phím chức năng cơ bản cần biết:

Phím tắt	Ý nghĩa
F6	Biên dịch chương trình
F5	Chạy chương trình
Ctrl+F5	Chạy chương trình (chế độ dừng CT sau khi chạy xong)
Shift+F5	Dừng chương trình
F9	Tạo/Hủy một điểm dừng (breakpoint) để chạy debug
F10	Chạy từng dòng lệnh (lướt qua hàm)
F11	Chạy từng dòng lệnh (đi vào qua hàm)

Ví dụ chương trình C

❖ Xuất ra màn hình dòng chữ: “Xin chào”

```
#include <stdio.h>
#include <conio.h>

void main()
{
    printf("Xin chào ");
    getch();
}
```

Hoặc

```
#include <stdio.h>
#include <conio.h>

int main()
{
    printf("Xin chào ");
    getch();
    return 0;
}
```

Tập ký tự trong C

❖ Tập ký tự trong ngôn ngữ C bao gồm các ký tự sau:

- 26 chữ cái latin lớn $A, B, C \dots Z$
- 26 chữ cái latin nhỏ $a, b, c \dots z$.
- 10 chữ số thập phân $0, 1, 2 \dots 9$.
- Các ký hiệu toán học: $+, -, *, /, =, <, >, (,)$
- Các ký hiệu đặc biệt: $::, ;, ", ', _ @ \# \$! ^ [] \{ \} \dots$
- Dấu cách hay khoảng trống.

❖ **Phân biệt chữ in hoa và in thường**

❖ Không được dùng các ký hiệu như: $\alpha, \varphi, \Omega \dots$

Các từ khóa trong C

- ❖ Từ khóa là các từ dành riêng của C.
- ❖ Ta không được dùng từ khóa để đặt cho các tên riêng.
- ❖ Các từ khoá thông dụng: int, char, float, long, double, if, else, switch, case, while, do, for, return, break, struct, unsigned, void,...

Tên và cách đặt tên

- ❖ Tên (identifier): dùng để xác định các đại lượng khác nhau trong một chương trình. Chúng ta có **tên hằng**, **tên biến**, **tên mảng**, **tên hàm**, **tên con trỏ**...
- ❖ Tên là **các kí tự đứng liền nhau**, gồm:
 - + *Các chữ cái*: **A,..., Z, a,..., z**
 - + *Các chữ số*: **0,..., 9**
 - + *Dấu gạch dưới*: **_**
- ❖ Có 2 loại:
 - **Tên chuẩn**: là tên do C đặt sẵn như tên kiểu: **int, char, float,...**; tên hàm: **sin, cos...**
 - **Tên do người lập trình tự đặt**.

Tên và cách đặt tên (tt)

❖ Quy tắc đặt tên:

- + *Không bắt đầu bằng số.*
- + *Không chứa dấu cách trắng, dấu chấm câu.*
- + *Không trùng với từ khoá.*
- + *Độ dài tối đa 32 kí tự*

❖ Ví dụ đặt tên đúng: a_1 , delta , x1 , _step ,
GAMA

❖ Đặt tên sai: 3MN, m#2, f(x), do

Cặp dấu chú thích (comment)

```
#include <stdio.h>
#include <conio.h>
int main () {
    char ten[50];          /* khai bao bien ten
                           kieu char 50 ky tu */
    printf("Xin cho biet ten cua ban !");
    scanf("%s",ten); /*Doc vao 1 chuoai la ten ban*/
    printf("Xin chao ban %s\n ",ten);
    //Dung chuong trinh, cho go phim
    getch();
    return 0;
}
```

- ❖ Khi biên dịch các phần chú thích bị bỏ qua
- ❖ Dùng **/*** và ***/**: chú thích dài nhiều dòng
- ❖ Dùng **//**: chú thích chỉ 1 dòng

Câu lệnh, khối lệnh và dấu chấm câu

❖ **Lệnh:** Thực hiện một chức năng nào đó (khai báo, gán, xuất, nhập,...) và được kết thúc bằng dấu chấm phẩy (;)

Ví dụ: `int x =5;`

❖ **Khối lệnh:** Gồm nhiều lệnh và được đặt trong cặp dấu ngoặc { }

❖ Trong chương trình có nhiều chỗ không dùng đến dấu ; vì đó **không phải câu lệnh**

Ví dụ: `#include <conio.h>`

Cấu trúc chương trình

```
#include <stdio.h>
#include <conio.h>
```

```
void main()
{
    printf("Xin chao ");
    getch();
}
```

```
//Khai báo thư viện
//Khai báo thư viện
```

```
//Hàm chính bắt buộc
//Bắt đầu
//In ra màn hình
//Dừng và chờ
//Kết thúc
```

❖ Lưu ý:

- *Dấu chấm phẩy*
- *Thứ tự lệnh*
- *Dấu bao chuỗi*

Các thành phần của chương trình C đơn giản

(1) **#include <Tên tập tin>**

hay **#include “Tên đường dẫn”**

Yêu cầu trình biên dịch đọc tập tin thư viện
mà chương trình dùng.

VD: `#include <stdio.h>`

`#include “math.h”`

`#include “C:\\BC5\\BIN\\math.h”`

Các tập tin thư viện thông dụng

- ❖ **stdio.h**: Tập tin định nghĩa các hàm vào/ra chuẩn (standard input/output). Gồm các hàm in dữ liệu **printf()**, nhập giá trị cho biến **scanf()**, xóa vùng đệm bàn phím **fflush()**....
- ❖ **conio.h** : Tập tin định nghĩa các hàm vào ra trong chế độ DOS (DOS console). Gồm các hàm xóa màn hình **clrscr()**, dừng màn hình nhập ký tự **getch()**...
- ❖ **math.h**: Tập tin định nghĩa các hàm tính toán gồm các hàm **abs()**, **sqrt()**, **pow()**, ...
- ❖ **alloc.h**: Tập tin định nghĩa các hàm liên quan đến việc quản lý bộ nhớ. Gồm các hàm **calloc()**, **realloc()**, **malloc()**, **free()**...

Các thành phần của chương trình C đơn giản (tt)

- (2) Hàm chính, là thành phần buộc phải có trong mọi chương trình C.
- **Dạng đơn giản:**

```
void main() { }
```

```
int main() { return 0; }
```

```
main() { return 0; }
```

Các thành phần của chương trình

C đơn giản (tt)

(3) Định nghĩa dữ liệu và các phát biểu.

- Các là phần thực thi của chương trình.
- Các phát biểu được đặt giữa cặp ngoặc { và } của hàm (main), tạo nên “*thân hàm*”.
- Mỗi *phát biểu (câu lệnh)* được kết thúc bởi ‘;’
- Các phát biểu cùng được đặt giữa { và } tạo thành *phát biểu ghép* (còn gọi *khối lệnh*).

Các bước để thực thi một chương trình

❖ Gõ và đặt tên tập tin: **hello.cpp**

```
#include <stdio.h>
#include <conio.h>

void main()
{
    printf("Xin chao cac ban");
    getch();
}
```

❖ Chạy chương trình với BC++

❖ Phân biệt: **F9** và **Ctrl+F9**

Một số qui tắc khi viết chương trình

- ❖ Mỗi câu lệnh có thể viết trên một hay nhiều dòng nhưng phải kết thúc bằng dấu “;”
- ❖ Chú thích có thể được viết trên một dòng, nhiều dòng hoặc trên phần còn lại của câu lệnh.
- ❖ Trước khi sử dụng một hàm chuẩn cần biết nó nằm trong thư viện nào và dùng toán tử **#include** để gắn tập tin đó vào đầu chương trình.
- ❖ Ví dụ: `#include <conio.h>`
- ❖ Một chương trình có thể **chỉ có một hàm chính** (**main**) hoặc có thể có thêm vài hàm khác.

Ví dụ

❖ Nhập vào 2 số, in ra tổng của chúng

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, tong;           //khai bao bien a, b, tong kieu so nguyen
    printf("Nhap so a: ");    //xuat ra man hinh
    scanf("%d",&a);           //nhap du lieu cho bien a
    printf("Nhap so b: ");    //xuat ra man hinh
    scanf("%d",&b);           //nhap du lieu cho bien b
    tong = a + b;             //tinh tong a va b
    printf("Tong a+b=%d ",tong); //xuat ra ket qua tong
    getch();                  //dung man hinh xem ket qua
}
```